Teil III der Vorlesung Objektorientierte Anforderungs- und System-Analyse (OOA)
30) Überblick über die OOA

Prof. Dr. Uwe Aßmann
Institut für Software- und Multimediatechnik
Lehrstuhl Softwaretechnologie
Fakultät für Informatik
TU Dresden
Version 20-0.3, 20.06.20

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Obligatorische Literatur

- Zuser, Kap. 7-9
- Störrle, Kap. 5

- Manfred Broy und Andreas Rausch. Das neue V-Modell® XT. Ein anpassbares Modell für Software und System Engineering. Informatik-Spektrum. Springer Berlin / Heidelberg. Volume 28, Number 3 / June, 2005, Pages 220-229 http://www.springerlink.com/content/l173638386334305/
- M. Kim et al. Service Robots for the Elderly. IEEE Explore. http://dx.doi.org/10.1109/MRA.2008.931636. UML/COMET software modeling method to use refinement to develop service robot software.
  - Uses use case diagrams, context diagrams, communication diagrams, statecharts.

Wer hat schon Java compiliert und ausgeführt?
Wer hat schon INLOOP genutzt?

Das Ziel des Studiums

- Muhammad Yunus, founder of Grameen Bank for microcredits in Bangla Desh
  http://muhammadyunus.org/
  - Social Business. Von der Vision zur Tat. Hanser, München 2010 (Originaltitel: Building Social Business, übersetzt von Werner Roller), ISBN 978-3-446-42351-0
  - https://de.wikipedia.org/wiki/Social_Entrepreneurship
  - http://de.wikipedia.org/wiki/Social_Business
  - http://socialbusinesspedia.com/
  - https://en.wikipedia.org/wiki/Grameen_family_of_organizations

> One of the most efficient teachings life has given to me is the insight that Human Beings possess an awesome creational and entrepreneurial potential.
> [M. Yunus, Social Business.]

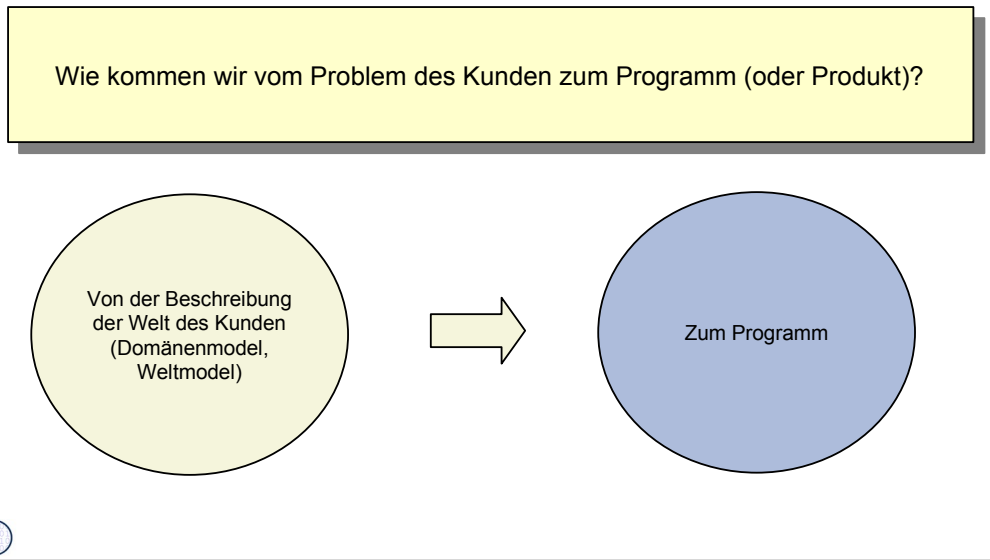
Das Ziel des Studiums (II)

- „Die Grameen-Bank ermuntert die Kinder ihrer Kreditnehmer auch zum Schulbesuch. .. Derzeit studieren mehr als 50000 Studentinnen und Studenten mithilfe von Ausbildungskrediten der Grameen-Bank...
- Wir ermuntern diese jungen Leute, sich fest vorzunehmen, dass sie sich niemals als Arbeitssuchende auf den Arbeitsmarkt begeben werden. Sie sollen später einmal Arbeitsplätze schaffen, nicht sich um Arbeit bewerben. Wir sagen ihnen: Euren Müttern gehört eine große Bank, die Grameen-Bank. Die hat einen Haufen Geld, mit dem sich jedes Unternehmen eurer Wahl auf den Weg bringen lässt. ***Warum wollt Ihr Zeit mit Arbeitssuche vergeuden, um dann für jemand anderen zu arbeiten? Werdet lieber Arbeitgeber, keine Arbeitnehmer.***
- ***Die Grameen-Bank ermutigt die Menschen von Bangladesh zur unternehmerischen Selbständigkeit und wirtschaftlichen Unabhängigkeit – weg von der Abhängigkeit.*"**
- Mohammad Yunus – Social Business. Von der Vision zur Tat. Hanser 2010.

We encourage the students who receive a study grant from Grameen:
It is your chance to create employment with your education.
Think about becoming an entrepreneur who uses his education to create jobs for others.
[M. Yunus, Social Business]

30.1 Überblick über die Objektorientierte Analyse

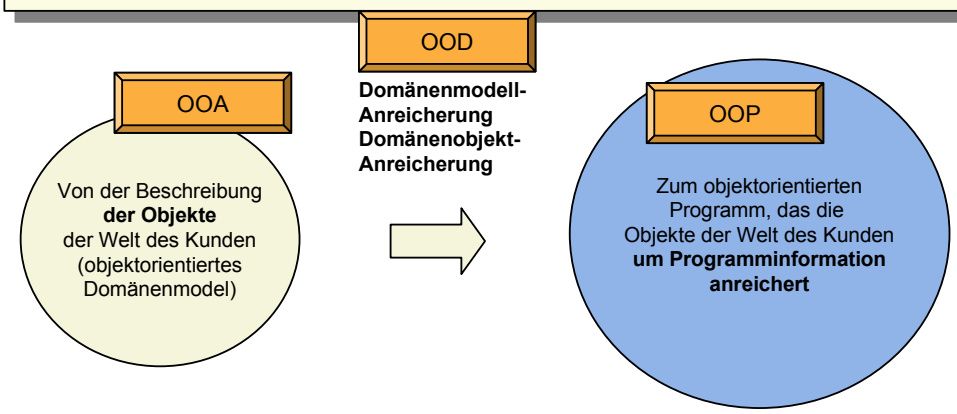
Die zentralen Frage der Softwaretechnologie



Diese Folie kennen wir schon. Sie erinnert uns an die Strategie der objektorientierten Programmentwicklung: von der Welt des Kunden aus, die im Domänenmodell (fachlichen Modell) erfasst wird, arbeiten wir uns durch Anreicherung von Details zum Programm vor.

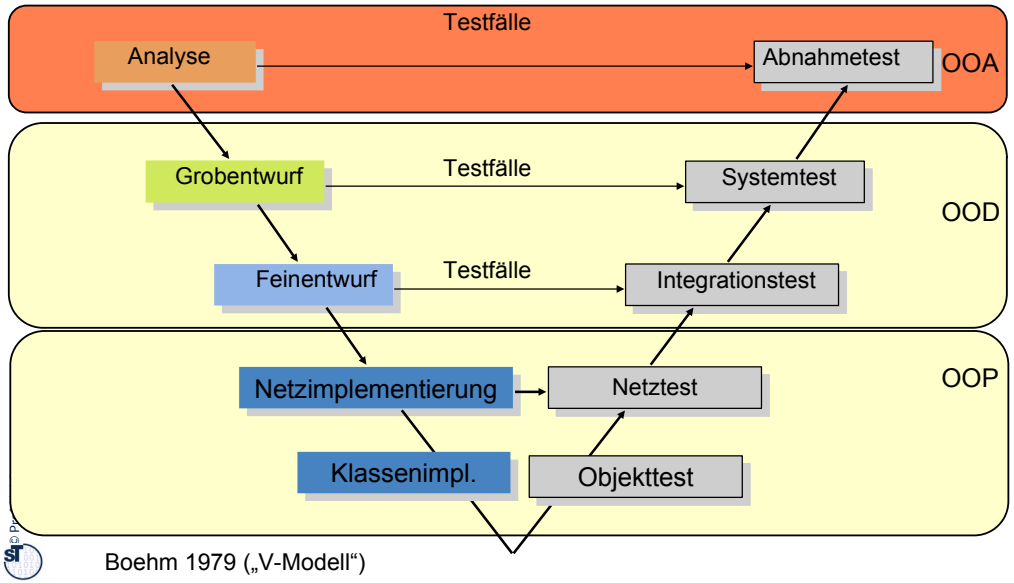
Die zentralen Fragen des objektorientierten Ansatzes

Wie kommen wir vom Problem des Kunden zum Programm (oder Produkt)?



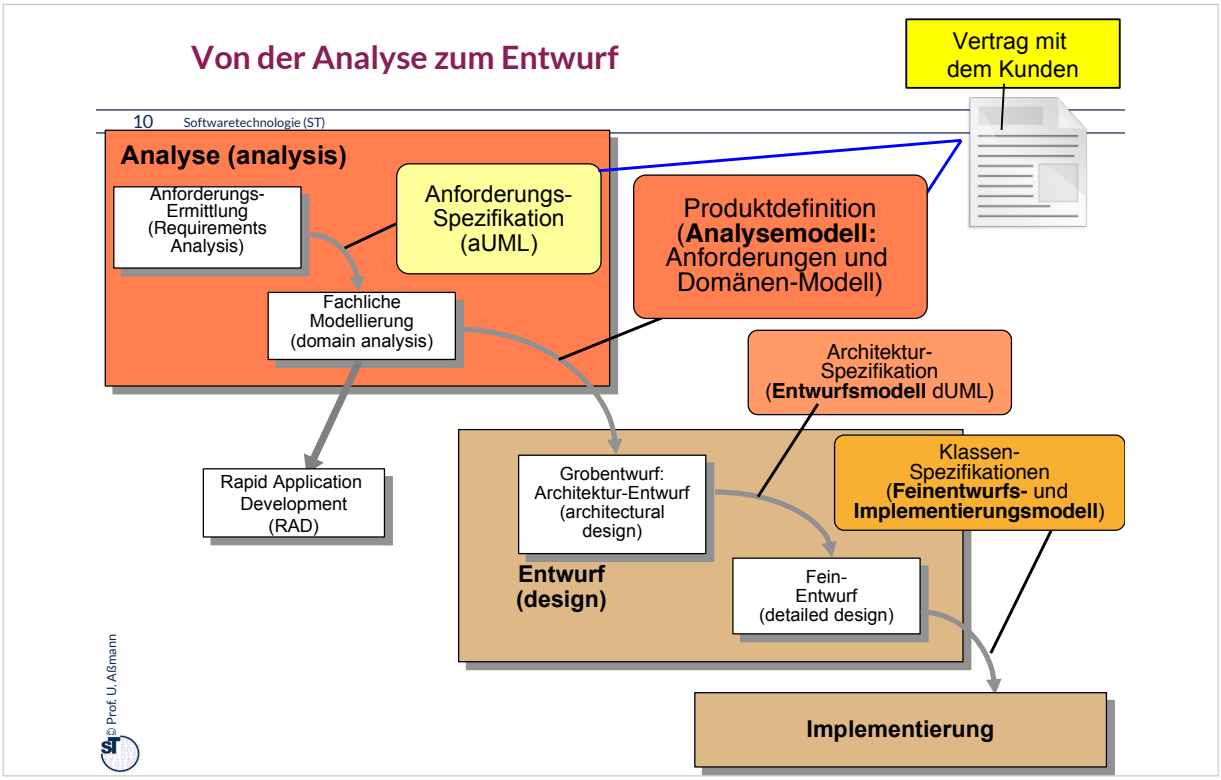
Anreicherung/Verfettung: Anreicherung durch technische Programminformation
„object fattening“: Anreicherung von Objekten des Domänenmodells

Softwareentwicklung im V-Modell



Boehm 1979 („V-Modell“)

Quelle fuer das Boehm-Zitat: Hesse/Merbeth/Frölich S. 37



•Die Arbeitsphase **Analyse** ermittelt, was der Benutzer vom System benötigt, das **Analysemodell** in aUML

- *Fachliche Analyse:* Begriffe und Objekte der Anwendungsdomäne (Domänenmodell, fachliches Modell)
- *Anforderungsanalyse:* Formulierung der Anforderungen
- *Systemanalyse:* Schnittstellen des Systems (Kontextmodell mit Funktionen, Daten, Klassen, Code)

•Die Arbeitsphase **Entwurf** ermittelt, was der Entwickler zusätzlich zu den Ergebnissen der Analyse ins System aufnehmen muss, was aber der Benutzer nicht sehen muss: das **Entwurfsmodell** in dUML

- Der **Grobentwurf** entwickelt die Architektur (“Programmieren im Großen”) im **Architekturmodell**
- Der **Feinentwurf** entwickelt die Struktur von Klassen oder Subsystemen (**Feinentwurfsmodell**)

•Die Arbeitsphase **Implementierung** vervollständigt und konkretisiert den Entwurf

- zum **Implementierungsmodell** (partielle Implementierung in jUML)
- dann durch Ergänzung zum abläuffähigen (Java-)Programm
- Klärt die Plattformabhängigkeiten des Programms

•

Modellierung, Spezifikation vs. Realisierung

- Zuerst das Problem verstehen (Problemanalyse),

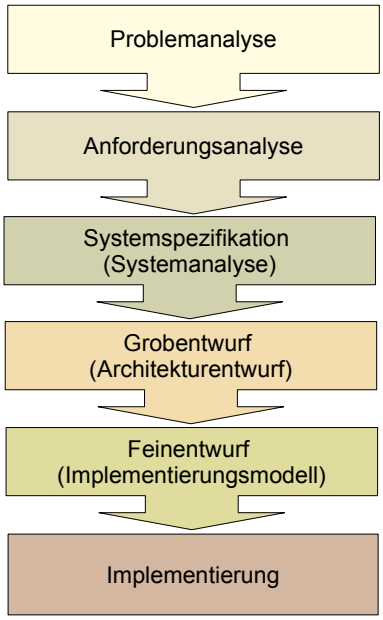
  dann Anforderungen festlegen (Anforderungsanalyse),

  und Funktionalität spezifizieren (Systemanalyse und -spezifikation)

  dann Architektur festlegen (Entwurf),

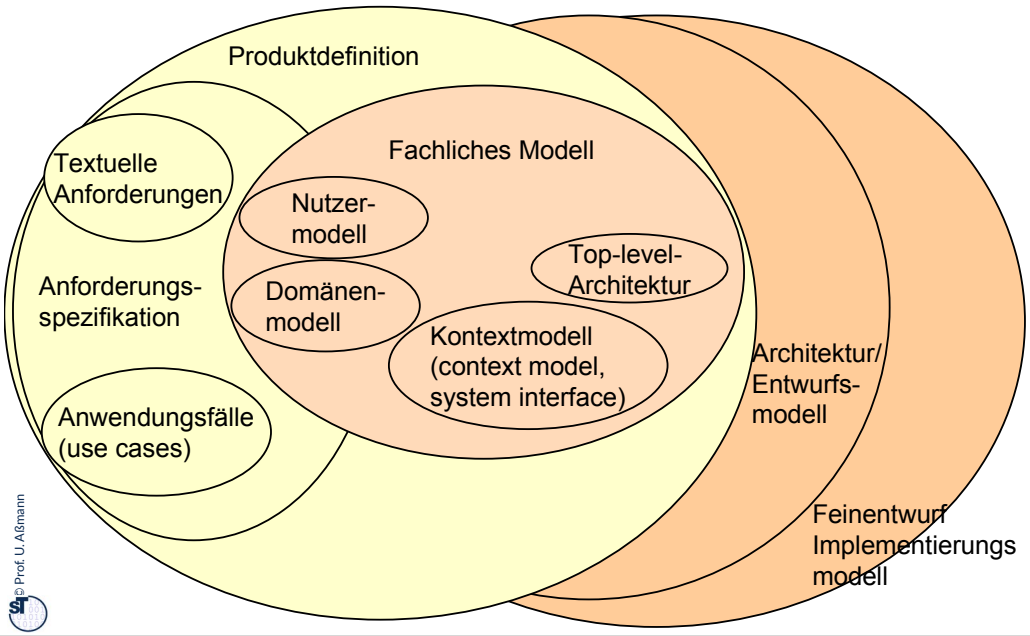
  dann in Implementierungsmodell überführen (Feinentwurf),

  und zuletzt alle Details festlegen (Implementierung)

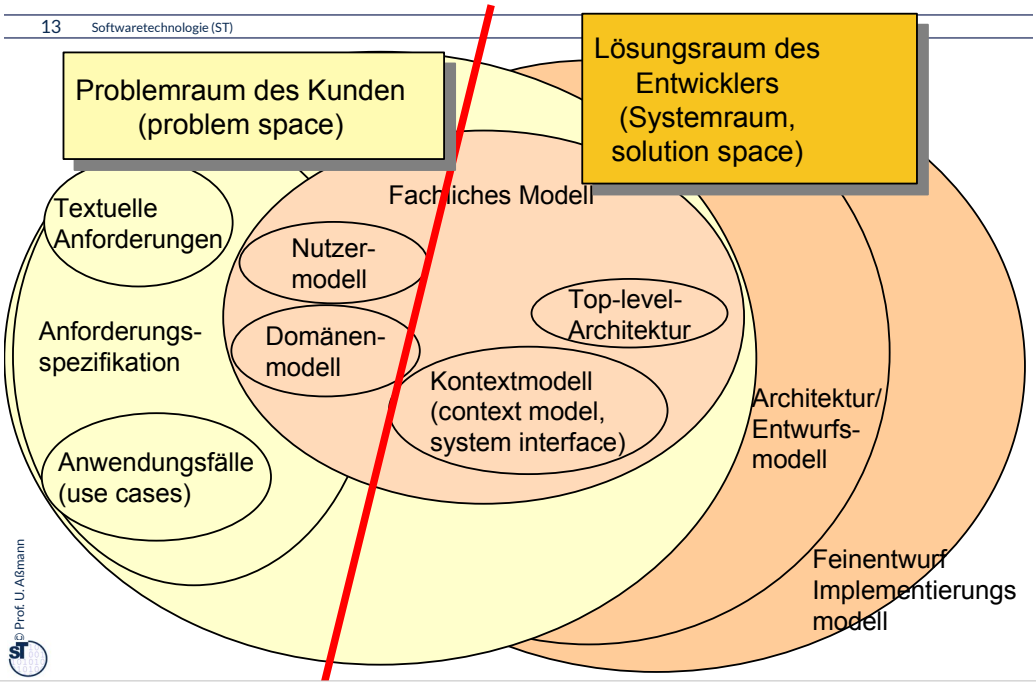


- Domänenmodelle (Fachliche Modelle) können im Sinne eines Prototyps bereits in Skriptprogrammiersprache realisiert werden (RAD, rapid application development, prototyping):
  - Keine Benutzeroberfläche, keine Datenhaltung
  - Eingeschränkte Funktionalität

Artefakte und Modelle im Prozess von den Anforderungen zum Feinentwurf



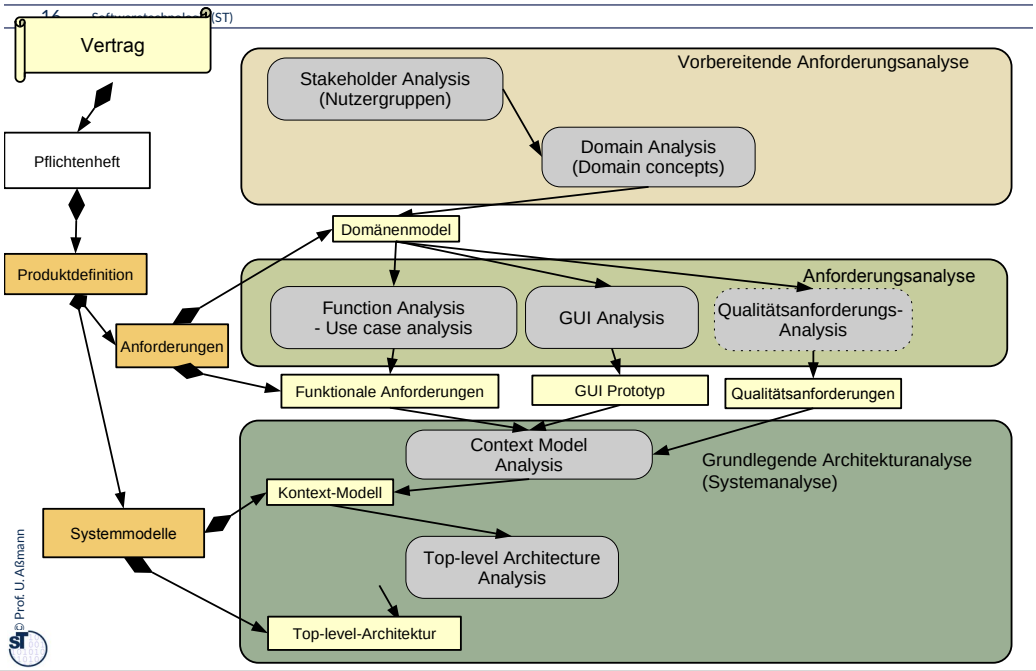
Artefakte und Modelle im Prozess von den Anforderungen zum Feinentwurf



Q5: Schritte der Modellierung in Bezug auf Schichten des Systems

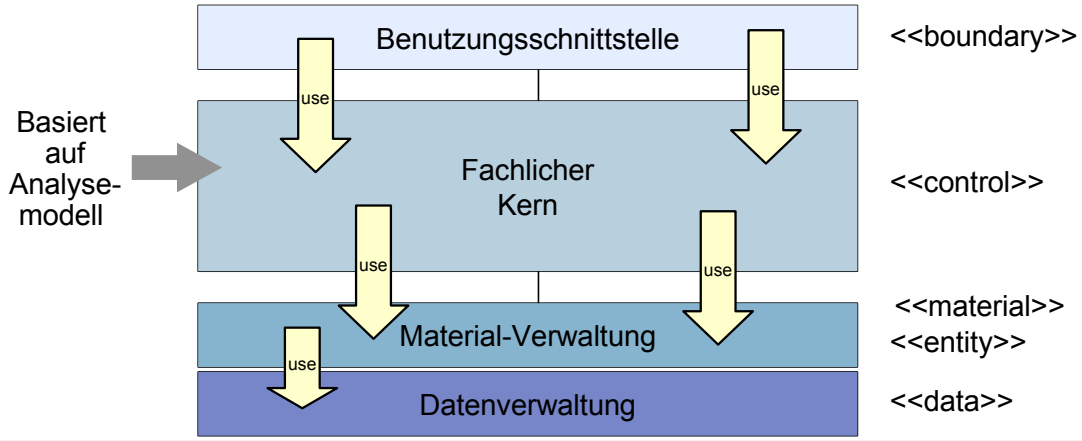
	Schichten	Analyse	Entwurf	Feinentwurf	Implementierung
		Verfeinerung			
Benutzungsschnittstelle (Boundary)	GUI				
	Controller				
Anwendungslogik (Control)	Kontextmodell	Aufstellung aus Domänenmodell; Erarbeitung System-schnittstellen	stabil	Umsetzen auf jUML	stabil
	Top-Level-Architektur	Verfeinerung des Kontextmodells	stabil	Umsetzen auf jUML	stabil
	Architektur		Ausarbeitung Architektur (PSM)	Einziehen von Plattformabhängigkeiten (PSM); Umsetzen auf jUML; Sequentialisierung	Details ausfüllen, Methoden ausprogrammieren
	Tools		Ausarbeitung Tools	Einziehen von Plattformabhängigkeiten (PSM); Umsetzen auf jUML; Sequentialisierung	Details ausfüllen, Methoden ausprogrammieren
Datenhaltung (Database)	Material	Aufstellung aus Domänenmodell		Einziehen von Plattformabhängigkeiten (PSM); Umsetzen auf jUML;	Details ausfüllen, Methoden ausprogrammieren

Q10: Drei-Schritt der Analyse (Anforderungen und fachliches Modell)



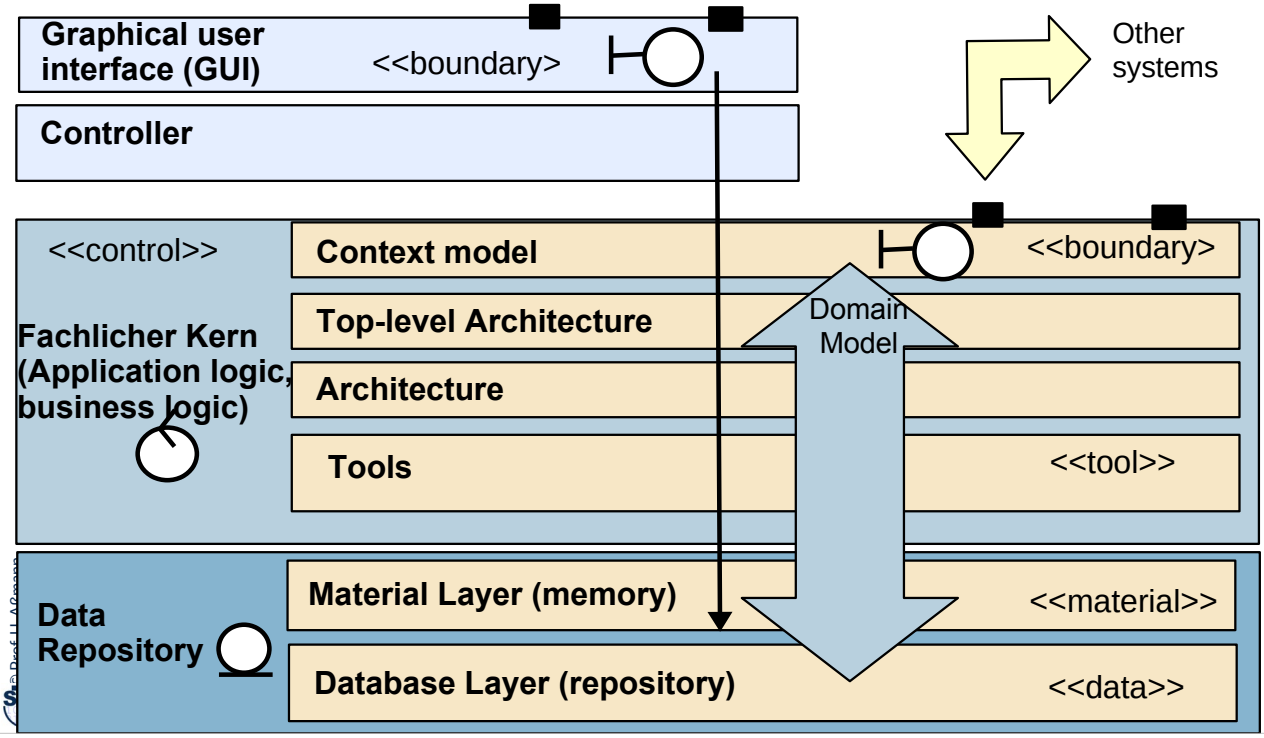
Architekturstil für Interaktive Anwendungen: Vier-Schichten-Architektur (BCED)

- Klassische Struktur eines interaktiven Anwendungssystems
- Integrationstest verläuft wegen azyklischer Benutzungsrelation (use) bottom-up: erst D, dann ED, dann CDE, dann BCED
- Fachlicher Kern (Anwendungslogik) kann weitere Schichten enthalten
  - Oft kapselt eine Facade eine Schicht nach oben ab, dann existieren bereits zwei Teil-Schichten



Q7: Verfeinerte BCED-Schichtung eines Systems

- Im Teil III+IV verwenden wir 7 Schichten in 3 Gruppen:



Statische und dynamische Aspekte der Modellierung

	Statisch	Dynamisch	Mittel der UML
Punktweise Modellierung	Klassen- bzw. Objektmodellierung Schichten und Architektur		Objekt-zentriert Metamodell-getrieben, Canvas-getrieben
		Lebenszyklen Parallele Prozesse	Aktionsdiagramme
Scheiben-(Schnitt-) Modellierung (slice modeling)	Relationale Modellierung (Netzmodellierung), Konnektoren		Relationen-zentriert Assoziationen Kollaborationen (Teams)
		Scheiben-Modellierung (Querschneidende Modellierung)	Szenarien-getrieben Interaktionsdiagramme

Zum Praktikum

- mehr als 95% aller Themen im Praktikum sind BCD-Architekturen
  - Oft mit Web-GUI
  - Unterscheidung zwischen GUI, Anwendungslogik und Datenhaltung ist essentiell
- Man sollte verstehen, dass aus dem Domänenmodell
  - die Datenhaltung folgt
  - die Typen der Daten folgen, die im Kontextmodell und in der Top-Level-Architektur fliessen
  - der GUI-Prototyp stark bestimmt wird (Kommunikation mit dem Benutzer)
- Die Entwickler oft nur für B, C, oder D zuständig sind

Überblick Teil III: Objektorientierte Analyse (OOA)

1. Überblick Objektorientierte Analyse
    1. (schon gehabt:) Strukturelle Modellierung mit CRC-Karten
2. Strukturelle metamodellgetriebene Modellierung mit UML
    1. Strukturelle metamodellgetriebene Modellierung für das Domänenmodell
    2. Strukturelle Modellierung von komplexen Objekten
    3. Strukturelle Modellierung für Kontextmodell und Top-Level-Architektur
3. Analyse von funktionalen Anforderungen (Verhaltensanalyse)
    1. Funktionale Verfeinerung: Dynamische Modellierung und Szenarienanalyse mit Aktionsdiagrammen
    2. Funktionale querschneidende Verfeinerung: Szenarienanalyse mit Anwendungsfällen, Kollaborationen und Interaktionsdiagrammen
    3. (Funktionale querschneidende Verfeinerung für komplexe Objekte)
4. Beispiel Fallstudie EU-Rent

End

- Wieso ist eine Anforderungsanalyse so wichtig?
- Erklären Sie die Unterschiede der drei wesentlichen Analyseschritte.
- Einige Folien sind eine überarbeitete Version aus der Vorlesung Softwaretechnologie von © Prof. H. Hussmann, used by permission.

30.A.1 Dokumente der Anforderungsanalyse

Inhalte eines Vertrags mit einem Kunden

- Pflichtenheft
  - Produktdefinition
    - Anforderungsspezifikation (das WAS)
      - Nutzermodell (stakeholders)
      - Domänenmodell
      - Funktionale Anforderungen
      - In SWT 2: Problemmodell, Zielmodell, Nicht-funktionale Anforderungen
    - Fachliches Modell (der Teil vom WIE, den der Kunde wissen muss)
      - Kontextmodell
      - GUI-Prototyp
      - Top-level-Architektur
  - Akzeptanztestfälle:
    - Messbare Akzeptanzkriterien, die bei der Abnahme vom Kunden abgehakt werden können. Ohne bestandenen Akzeptanztest keine Bezahlung!
- Preisliche Regelung
- *Achtung: In der Literatur wird der Begriff "Analysemodell" sowohl für die Produktdefinition als auch nur für das fachliche Modell verwendet!*

Inhalte der Anforderungspezifikation (WAS?)

- *Nutzermodell (stakeholder model):* Liste oder UML-Klassendiagramm aller am System Interessierten
  - In SWT-2 verfeinern wir das, in dem wir über die Ziele der stakeholder nachdenken
  - Enthält die Benutzer des Systems (die *Aktoren*)
- *Domänenmodell (domain model):*
  - Termini, Struktur und Grundkonzepte des Aufgabengebiets
  - Schaffung einheitlicher Terminologie für die Anforderungsspezifikation
  - Aus der Sicht des Kunden
  - Zusammenhang mit Anforderungsspezifikation sichern
  - Implementierungsaspekte ausklammern: Annahme *perfekter Technologie*
- *Problemmodell, Zielmodell* (s. SWT-2)

Inhalte der Anforderungspezifikation

- *Funktionale Anforderungen: Funktionale Essenz* des Systems. Was muss das System können?
  - Nicht das Wie, sondern nur das Was
  - möglichst quantitativ (z.B. Tabellenform)
  - eindeutig identifizierbar (Nummern)
  - Notation: Anwendungsfalldiagramme (Nutzfalldiagramme), Funktionsbäume oder textuell. Manchmal auch mathematisch
- *Nicht-funktionale Anforderungen (Qualitätsanforderungen)* (s. SWT-2)
  - Effizienzanforderungen
    - Resourcenausnutzung: Antwortzeit, Speicherbedarf, Last, Durchsatz, Energieverbrauch
  - Sicherheitskriterien
    - Zuverlässigkeit, Einbruchssicherheit, Privatsspährenschutz
  - HW/SW-Plattform
  - Entwicklungs- und Produkt-Standards

Inhalte des Fachlichen Modells (Fachkonzept) (Das WIE, das der Kunde wissen muss)

- *Kontextmodell*: äussere Schnittstellen des Systems
  - Ein- und Ausgabekanäle, Masken, Abfragen
  - Daten, die ein und aus fliessen, im Domänenmodell typisiert
- *GUI Prototyp:* Prototypische Masken, Formulare, Bildschirme, die den GUI ausmachen: Wie sieht das Programm aus?
- *Top-level Architektur (Initiale Architektur, Facharchitektur):* Bestimmt die Hauptkomponenten des Systems und ihre Interaktionen, ohne auf Details einzugehen
  - Verfeinert das Kontextmodell um eine Stufe, d.h. die top-level Architektur
  - Stellt das dar, was der Kunde von der Systemarchitektur wissen muss

Voller Ablauf der Analyse (s. SWT-2)